Regarding Migration and the Preserving Digital Information Paper
DMS 98-02-03/ Updated 98-03-04

This paper follows discussion of the "An Analysis of Information Migration" paper, dated 98-01-25, at the January 1998 US archiving workshop. It attempts to address some issues raised there, as well as some issues raised in that paper. To provide additional context, I have taken another look at the Preserving Digital Information (PDI) paper, and I urge all of us to give it a second reading.

Issue: What is covered under the term "Migration" ?

With regard to Migration, the PDI paper clearly defines it to include not just refreshing media, but also format changes that may even 'clean up' extraneous noise. They refer to this as preserving digital integrity, which I find misleading because it sounds like 'bit preservation' when they are really addressing the integrity of the key information. Their approach clearly gets away from the simple view that only 'strict transmutation' (i.e., there is a one-to-one mapping between the resulting entities of the old representation and the resulting entities of the new representation) should be considered a part of migration (Lou's suggestion at the January US meeting). We are really just talking definitions, because all of these things need to be (and are) covered in the "Migration" section of version 2.0. Of course, the terms and definitions are the heart of a reference model, and are critical to its success. If we use "migration" to be this general case, as it is used in the PDI paper, we need to define some categories of migration. We are doing this with the 'Repackaging' concept, which includes media refreshing as one case. But when it comes to 'transmutations', I think we need more distinctions.

Issue: How do we distinguish new versions from new AIPs?

Lou has brought up this transmutation-related issue which I might characterize as: what types of changes don't result in a new version of an AIP, what types do, and when is a new AIP formed? My current thoughts are as follows:

1. Certain types of Repackaging, including Replication (i.e., media refreshing), don't require the assignment of a new version and also don't require updates to PDI (except possibly to Fixity?). It should be obvious in these cases that information has been preserved.

2. Types of Repackaging that also require changes to the Content Information to accommodate them, would require a new version. However the intent must clearly be to preserve the original content information as much as possible. A new version means that Provenance needs to be updated because the immediate source has been revised, in a sense, by the changes made. The chain of custody concept still makes sense. The new version is meant, from an archival perspective, to replace the previous version. The previous version may be kept as long as practical, but its deletion is not considered to be a true information loss. For physical media, this is like going from paper to film. (Bruce - what is the traditional archival view of this process?)

3. Transmutations (on the Content Information) would require a new version. The intent must be to preserve the original content information as much as possible, but it may include reducing 'extraneous noise' (e.g., reducing representation features not being used to convey source information - like going from a 32-bit real to a 5 character decimal representation). It may also include adding representation features for technical compatibility when they don't convey new information. This is viewed as case 2 above. Another example would be going from 8mm film to digital VHS to preserve home movies. If the intent was to get rid of the originals because it was felt that the VHS version was adequate to preserve the key information, then the VHS is a new version. Even though clearly some resolution, and therefore some information, has been lost, it has been decided that this is an adequate representation of the original.

4. Derived AIP which involves changes to the Content information, either to add or subtract in some substantial way, would result in a new AIP and NOT a new version. It is taken as a different product and in no way directly replaces the source AIP. An example would be producing an AIP with ASCII replacing real numbers when the ASCII format is known, or expected, to result in actual information loss but it also confers ease of use. Another example would be applying processing algorithms to produce a new product from the old. In this latter case, the original is one product, even if it includes the algorithms to be run. The actual running of the algorithms is a significant event, which needs to be documented as to when and by whom and in what environment this processing occurred. The result is a new product from an archival perspective. Another example is similar to the 8mm film to digital VHS example above, but in this case it is determined that the loss of resolution is not acceptable and thus the VHS form is considered a new product (for convenient access) that can not take the place of the 8mm film.

The distinction of case 4 from case 3 appears to be an archival judgement call, when it is not otherwise clear. So no firm line can be drawn.

Distinguishing between cases 1 and 2 needs additional analysis to make the boundaries between Packaging and Content clear.

Separating Content from Packaging, and Repackaging
DMS 98-02-05/ Updated 98-03-04

Simple Case

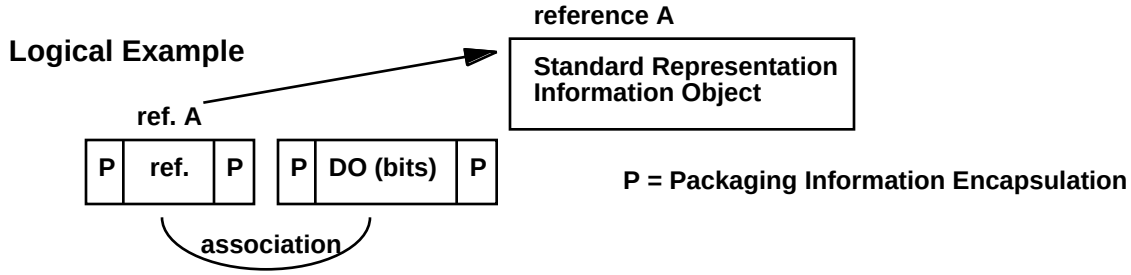
Consider the simplest case where the Content Information can be viewed as a single sequence of bits (object DO) together with a separate Representation Object (describing those bits). To understand this content information, we must know which of the two objects to begin with and how to understand this starting object. We don't want to start with the Representation Object because it may be applicable to, and used to, understand a number of different other objects. Therefore we want to start with the DO object and be told what information is to be used to understand the DO object. Therefore we need the

reference, or pointer, to the Representation object, and we need the Representation object if we don't already have it.

Therefore the most basic functions of packaging information appears to be the following:

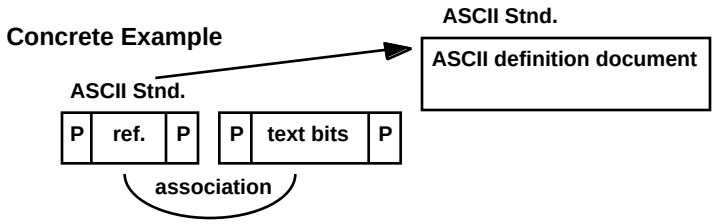
- o Identify and delimit the starting object
- o Provide an understandable identification (i.e., an understandable pointer) of a standard or other citable document to be used to understand the starting object

This is shown in the following logical example where a sequence of bits (DO) is delimited by packaging bits (P). DO is the starting object. Associated with this is Reference A which is also delimited by packaging bits. Reference A points to a standard representation object that can be used to understand the DO bits.



The 'understandable pointer', or reference identifier, must certainly be understandable to the archive, but it is most likely to be much more useful if it is also widely understandable to the designated communtity because this will reduce the need to either make a translation of the pointer when information is requested by the Consumer, or alternatively to chase down the pointer to retrieve the object pointed to and then provide it to the Consumer. The 'standard or other citable document' being 'pointed to' is actually part of the Content Information, but it is referenced by name only, either implicitly or explicitly. Here are some examples:

o Use ASCII to understand the file A's content (text bits): The citable reference is "ASCII" and the starting object is the content of file A. The reference identifier and the starting object are part of the Content Information. The archives packaging information delimits the starting object, delimits the reference identifier, associates the two, and thereby provides the link to the external ASCII document. The delimitation/packaging of this external referenced object is not considered to be a part of the packaging of the Content Information, by definition, even if a copy of the referenced object is maintained within the archive for convenience because the citable reference identifier is considered to be sufficient to identify the referenced document and lead to its acquisition. We can say that the 'Reference Identifier' stands in for the external part of the Content Information. Schematically this looks as follows:



- **Encapsulated bits, using archival standard packaging**
- **Globally understood pointer in archival standard syntax identifies representation information to start the understanding (e.g., Citable reference, or ISBN #)**
- **Pointer is associated with encapsulated bits**

o Use the ISBN 2430-3440... document to understand file A's content: The citable reference is "ISBN..." and the starting object is the content of file A. The packaging information and content information are as described in the example above.

o Use ASCII text-display software to understand file "README's" content, written in English: The citable reference is ASCII, the text-display software supports viewing data conforming to the ASCII standard, and the starting object is the content of file README. Note that simply incorporating a file with the name README suggests, implicitly, the above view and thus provides this basic packaging functionality implicitly. In this case, the ASCII display software substitutes for actually acquiring the ASCII standard. The packaging information and content information are as described in the two examples immediately above.

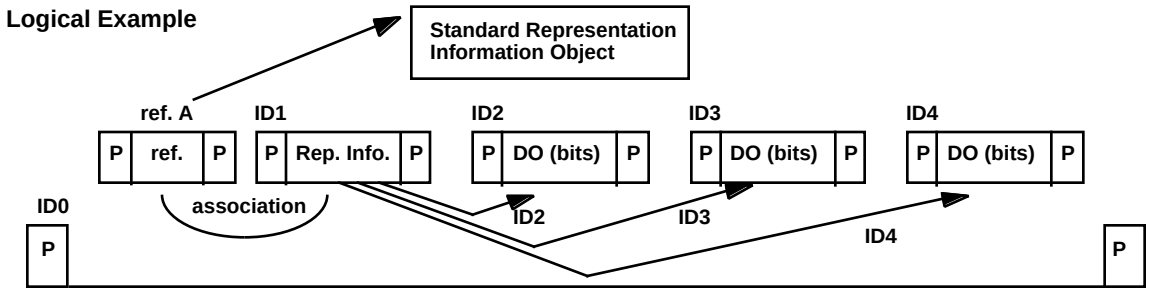
o On the surface of a CD-ROM is the statement "use CD-ROM/ISO-9660/ISO Packaging Standard xyz to understand the content of this media volume": The citable reference is "CD-ROM/ISO-9660/ISO Packaging Standard xyz " and the starting object is the media volume. Presumably the ISO Packaging Standard xyz will describe how to futher understand the content within ISO-9660. The implication of this example is that the starting object is all the bits on the CD-ROM, which is thus a part of the Content Information, and all the documents represented by "CD-ROM/ISO-9660/ISO Packaging Standard xyz " are also part of the Content Information. The packaging information is the physical media, which contains the bits and also carries the reference identifier on its disk surface, thereby linking the two.

Alternatively, a file on the CD-ROM could be identified as the starting file and the ISO Packaging Standard could be the referenced object. The packaging information would need to convey this information, as well as delimit the starting file using CD-ROM/ISO-9660 standards.

Typical Case

Now consider the more typical case where there are multiple objects, in addition to reference identifiers, to be packaged and related. Each of the objects must be delimited,

identified, and collected into an identified set. Each object must have associated representation information, and there must be representation information for the set as a whole to give the relationships among the objects. This is shown schematically in the following figure where individual object representation information for ID2, ID3, and ID4 may be considered to be included in ID1, or alternatively it is not shown to reduce diagram clutter.



In this example, ID1 has direct pointers to all the other packaged objects. An alternative example arrangement might have ID1 pointing to ID2, and ID2 describing and pointing to ID3 and ID4. Thus, in one way or another, ID1 directly or indirectly points to all the other packaged objects. Note that ID1, 2, 3, and 4, as identifiers associated with each object, are part of the packaging information. At the same time, the use of ID2, 3, and 4 within the content of ID1 is part of the Content Information. Thus the Content refers to packaging identifiers. This coupling of content to packaging can make migration difficult when the packaging identifiers need to change because then the Content Information must also be changed. If this happens, then the result of the migration is a new version. This can be avoided by constructing a standard packaging mapping-table that maps identifiers used inside content to packaging identifiers. It would be much easier to update this standard content-to-packaging interface object (i.e., the mapping table) than to update the content information during a migration in which packaging identifiers needed to change. Under such a migration the version would not need to change because only the packaging information changed.

To summarize, at minimum the packaging information in the general case must:

o Encapsulate and provide an Identifier for each separate object
o Encapuslate the set of objects and provide an Identifier for the set
o Identify a starting object and associate a reference identifier to a standard represention information object. The starting object must address (point to), either directly or indirectly, each of the other packaged objects.
o If needed for other than the starting object, package and associate a reference identifier, pointing to a standard representation information object, for each object packaged.

Interim Summary

At this point, the paper has clarified a scope for the function of migration and has clarified

the role of packaging information as opposed to Content Information. At their root they stem from a different set of bits when they are digitally based, or are physically distinct when physically based. However there is a minimum set of interactions between the two that exists and must be supported by the implementations, as described in the general packaging case above.

If a migration can be accomplished without needing to change content information, and at most only changing packaging information, then there is no version change. However if the content information is changed, and the intent is that the new content information is intended as an archival replacement of the previous form, then a migration has taken place and the new form is considered to be a new version of a previous AIP. Alternatively, if the new content information is NOT considered to be an archival replacement of the previous form, then migration has NOT taken place and the new form is considered to be a new product which can, of course, become a new AIP.

At this point, the role of PDI within packaging information has been assumed to be 'just more content to be packaged.' This needs to be looked at more closely. For example, can fixity be maintained when packaging changes, in all cases, or is this like content information in that some packaging changes do, and some don't, impact fixity? What about the PDI Reference for the Content Information? How is it related to Content and Packaging Information, and when does it need to change or be extended?